

PERBANDINGAN KECEPATAN Pencarian Data SQL dan NOSQL

Muhammad Firdaus Abdi, Agung Susanto, Kusnawi

Magister Teknik Informatika, Universitas Amikom, Yogyakarta

Jl. Ring Road Utara, Kabupaten Sleman, Daerah Istimewa Yogyakarta

mfirdaus.abdi29@gmail.com, agung12022170@gmail.com, kusnawi@amikom.ac.id

Abstract - The development of database technology requires a quality of data, where a database is needed in various applications that support the performance of a company or office, so that the database is said to be good, it must be managed appropriately, efficiently in terms of speed or time. For this reason, the use of this database is still the *prima donna* used in running a system. In this study, testing a database to determine the query response time from a Structured Query Language (SQL) and NoSQL (Non-SQL) system which was tested with a sales or sales database where the data records were (100,500,1000,100000, 2279879). Testing focused on data search includes queries displaying all data, one of the data, AND, OR, AND and OR, Order By (asc), and Order By (desc). The purpose of this research is to produce a comparative analysis of the query response time between SQL and NoSQL so that it is known which database is superior in a data search for use. The results of this study indicate that NoSQL is superior in searching for a database, but there are certain NoSQL searches that are slower than the SQL search process.

Keywords - NoSQL, SQL, Response Time, Search Data, Comparison Base Data.

Abstrak - Perkembangan teknologi basis data menuntut sebuah kualitas sebuah data, yang dimana sebuah basis data sangat diperlukan di berbagai aplikasi penunjang kinerja sebuah perusahaan atau kantor, untuk itu agar basis data dikatakan baik harus dikelola dengan tepat, efisien dari segi kecepatan atau waktu. Untuk itu penggunaan basis data ini masih menjadi primadona digunakan dalam menjalankan sebuah sistem. Pada penelitian ini melakukan pengujian sebuah database untuk menentukan waktu query response dari sebuah sistem Structured Query Language (SQL) dan NoSQL(Non SQL) yang diujikan dengan database sales atau penjualan yang dimana record data (100,500,1000,100000, 2279879). Pengujian terfokus pada pencarian data meliputi query tampil seluruh data, salah satu data, AND, OR, AND dan OR, Order By (asc), dan Order By (desc). Tujuan penelitian ini menghasilkan analisis perbandingan waktu query response antara SQL dan NoSQL agar diketahui basis data yang lebih unggul dalam sebuah pencarian data untuk digunakan. Hasil penelitian ini didapat bahwa NoSQL lebih unggul dalam pencarian sebuah data namun ada pencarian tertentu NoSQL lebih lambat dari proses pencarian SQL.

Kata Kunci - NoSQL,SQL, Response Time,Pecarian Data,Perbandingan Basis Data.

I. PENDAHULUAN

Saat ini berbagai teknologi basis data yang digunakan untuk memudahkan penggunaanya dalam mengakses data[1], yang dimana seiring perkembangan waktu data informasi yang masuk kedalam basis data mengalami peningkatan yang besar[2], peningkatan data yang besar dapat mempengaruhi kecepatan akses pada database[3]. Database relasional menjadi andalan dalam beberapa dekade terakhir dengan Structured Query Language (SQL) untuk melakukan pengolahan data[4] juga SQL sendiri memiliki portabilitas yang tinggi karena sistem bahasa yang terstruktur[5].

Disisi lain timbul perkembangan dalam mengelola database yaitu NoSQL(Non SQL)[6] dengan fitur yang dianggap lebih fleksibel dan skalabel secara horizontal[7], juga tidak menggunakan sistem penyimpanan berdasarkan tabel[8], namun menggunakan metode baru yaitu key-value, big table, document-oriented, dan graph[9]. NoSQL sendiri dianggap dapat mengatasi keterbatasan pada SQL oleh karena itu NoSQL dikenal sebagai non-relasional.[4].

Performa dari SQL dan NoSQL sendiri tentu berbeda[4] oleh karena itu dalam hal kecepatan pencarian data perlu diuji [10] untuk mengetahui perbedaan itu sendiri.

Hasil dari penelitian ini nantinya diharapkan dapat mengetahui manakah yang lebih cepat[11] dalam pencarian data skala besar antara SQL dan NoSQL[12]. Juga agar dari penelitian ini dapat diketahui akses mana yang lebih cepat untuk memproses pencarian sebuah data.

II. METODE PENELITIAN

Penelitian ini akan melakukan beberapa tahapan yaitu konfigurasi database pada SQL dan NoSQL, Perancangan, dan Pengujian.

A. Konfigurasi database pada SQL dan NoSQL

Tahap ini aplikasi SQL dan NoSQL sudah harus dipastikan ada pada komputer, selanjutnya data yang sama di masukkan pada SQL dan NoSQL.

B. Perancangan

1. Data yang digunakan
Data yang akan diuji pada penelitian ini yaitu pengujian *datasheet sales*/penjualan.
2. Percobaan
Data percobaan yang dilakukan yaitu 100,500,1000,100000, 2279879. *Query* yang dilakukan yaitu pencarian data.

C. Pengujian

Dari proses pencarian data kemudian dihitung waktu respon yang dihasilkan untuk menjadi tolak ukur perbandingan pencarian data SQL dan NoSQL.

3. HASIL DAN PEMBAHASAN

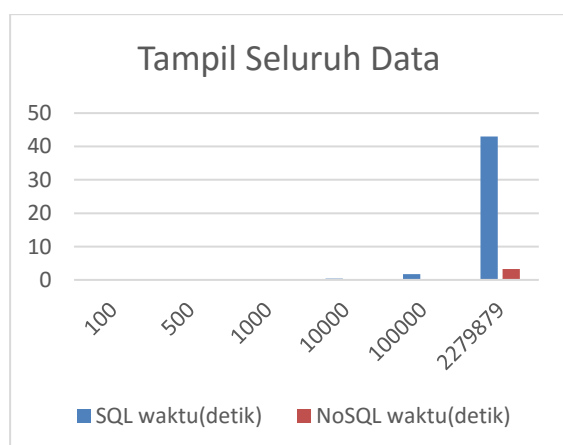
A. Hasil pengujian

1. Pengujian yang pertama pencarian data dengan menampilkan seluruh data.

Tabel 1. Pencarian menampilkan semua data

No.	Jenis	Sintaks	Jumlah Data	Waktu (detik)
1	SQL	SELECT * FROM sales.dbo.T100	100	0,075
			500	0,16
			1000	0,18
			10000	0,401
			100000	1,732
			2279879	42,949
2	NoSQL	db.T100.find().explain("executionStats")	100	0,001
			500	0,001
			1000	0,001
			10000	0,004
			100000	0,038
			2279879	3,307

Dari tabel pencarian dengan menampilkan seluruh data dari percobaan dengan jumlah data *record* yang berbeda ketika data itu tidak banyak SQL dan NoSQL tidak berbeda jauh, akan tetapi ketika data itu berjumlah besar maka terlihat perbedaan NoSQL lebih cepat mencari dan menampilkan seluruh data.



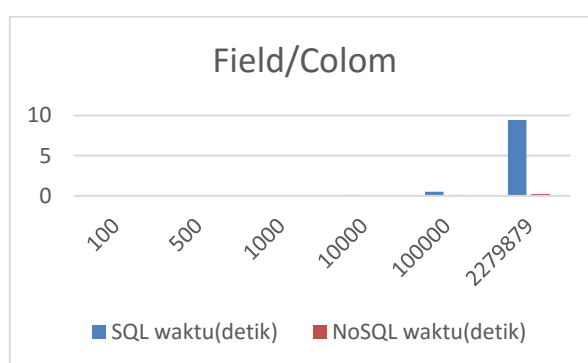
Grafik 1. Pencarian menampilkan seluruh data

2. Pengujian kedua pencarian data dengan menampilkan data per *field/kolom*.

Tabel 2. Pencarian menampilkan data per *field/kolom*

No.	Jenis	Sintaks	Jumlah Data	Waktu (detik)
1	SQL	SELECT Invoice_Item_Number FROM sales.dbo.T100	100	0,001
			500	0,054
			1000	0,064
			10000	0,104
			100000	0,514
			2279879	9,426
2	NoSQL	use sales; db.getCollection("T100").find({}, {"Invoice_Item_Number": "\$Invoice_Item_Number", "_id": NumberInt(1)}))explain("executionStats")	100	0,001
			500	0,001
			1000	0,001
			10000	0,02
			100000	0,131
			2279879	0,224

Pengujian kedua ini mencoba menampilkan data satu kolom saja, hasilnya juga sama ketika data berjumlah besar pencarian data lebih cepat dilakukan oleh NoSQL.



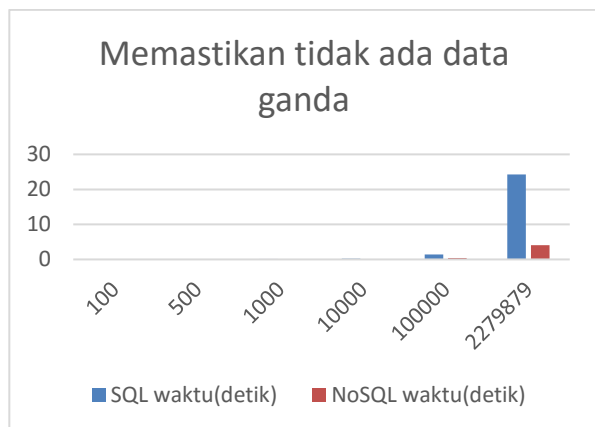
Grafik 2. Pencarian menampilkan data per *field/kolom*

3. Pengujian ketiga pencarian data untuk memastikan tidak ada data ganda pada data *index*.

Tabel 3. Pencarian untuk menampilkan data bertujuan memastikan tidak ada data ganda.

No.	Jenis	Sintaks	Jumlah Data	Waktu (detik)
1	SQL	SELECT DISTINCT Invoice_Item_Number FROM sales.dbo.T100;	100	0,001
			500	0,078
			1000	0,096
			10000	0,26
			100000	1,437
			2279879	24,280
2	NoSQL	db.getCollection("T100").explain("executionStats").aggregate([{"\$project": {"Invoice_Item_Number": "\$Invoice_Item_Number", "_id": NumberInt(1)}, {"\$group": {"_id": null, "distinct": {"\$addToSet": "\$\$ROOT"}}}],	100	0,001
			500	0,001
			1000	0,006
			10000	0,037
			100000	0,411
			2279879	4,033

Pengujian ketiga ini mencoba menampilkan data dan juga untuk memastikan tidak ada data ganda, hasilnya juga sama ketika data berjumlah besar pencarian data lebih cepat dilakukan oleh NoSQL.



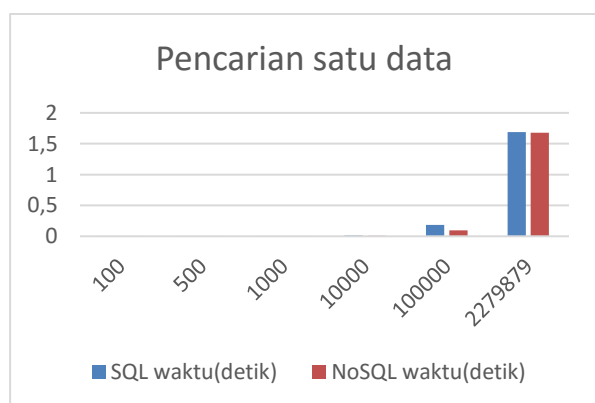
Grafik 3. Pencarian untuk menampilkan data bertujuan memastikan tidak ada data ganda.

4. Pengujian keempat pencarian salah satu data

Tabel 4. Pencarian untuk menampilkan salah satu data.

No.	Jenis	Sintaks	Jumlah Data	Waktu (detik)
1	SQL	SELECT * FROM sales.dbo.T100 WHERE City='SHELDON'	100	0,001
			500	0,001
			1000	0,001
			10000	0,012
			100000	0,186
			2279879	1,686
2	NoSQL	db.getCollection("T100").find({ "City" : "SHELDON" }).explain("executionStats");	100	0,001
			500	0,001
			1000	0,001
			10000	0,010
			100000	0,096
			2279879	1,680

Pengujian keempat ini mencoba menampilkan satu data pada database, hasilnya kecepatan pencarian SQL dan NoSQL tidak jauh berbeda meskipun pencarian data dengan jumlah yang besar.



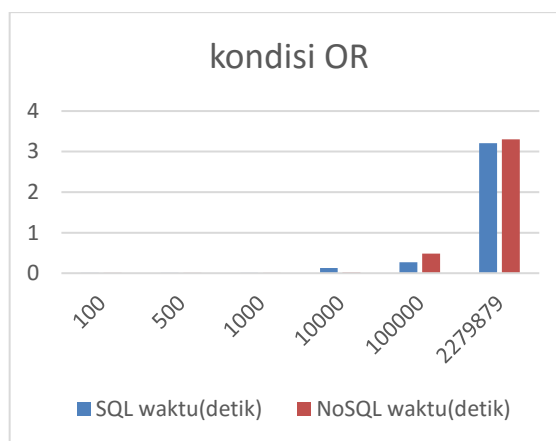
Grafik 4. Pencarian untuk menampilkan salah satu data.

5. Pengujian kelima pencarian dengan kondisi OR

Tabel 5. Pencarian kondisi OR

No.	Jenis	Sintaks	Jumlah Data	Waktu (detik)
1	SQL	SELECT * FROM sales.dbo.sales_2016 WHERE City='SHELDON' OR Zip_Code='52404';	100	0,001
			500	0,001
			1000	0,001
			10000	0,127
			100000	0,266
			2279879	3,206
2	NoSQL	db.getCollection("sales_2016").find({ "City" : "SHELDON", "Zip_Code" : "52404" }).explain("executionStats");	100	0,001
			500	0,001
			1000	0,001
			10000	0,016
			100000	0,482
			2279879	3,306

Pengujian kelima mencoba mencari salah satu data pada tabel city atau zip_code, jika salah satu terpehuni maka data akan tampil. Dan hasil pencarian ini waktu antara SQL dan NoSQL tidak berbeda jauh.



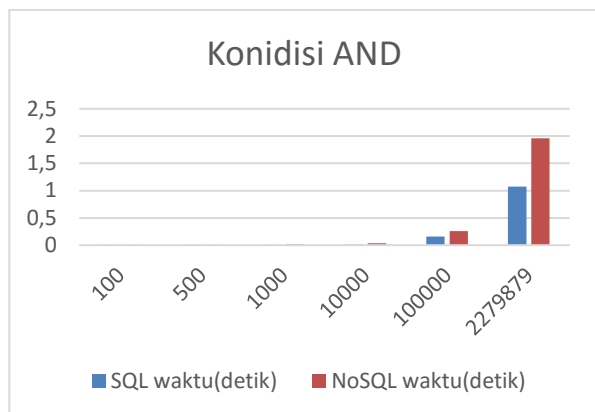
Grafik 5. Pencarian dengan kondisi OR

6. Pengujian keenam pencarian dengan kondisi AND

Tabel 6. Pencarian kondisi AND

No.	Jenis	Sintaks	Jumlah Data	Waktu (detik)
1	SQL	SELECT * FROM sales.dbo.sales_2016 WHERE City='SHELDON' AND Zip_Code='51201';	100	0,001
			500	0,001
			1000	0,001
			10000	0,013
			100000	0,159
			2279879	1,071
2	NoSQL	db.getCollection("sales_2016").find({ "City" : "SHELDON", "Zip_Code" : "51201" }).explain("executionStats");	100	0,001
			500	0,001
			1000	0,016
			10000	0,032
			100000	0,260
			2279879	1,962

Pengujian keenam mencoba mencari data pada tabel city atau zip_code, jika salah keduanya terpehuni maka data akan tampil. Dan hasil pencarian ini waktu antara SQL cepat ketika memproses pada data besar dan NoSQL lebih lambat.



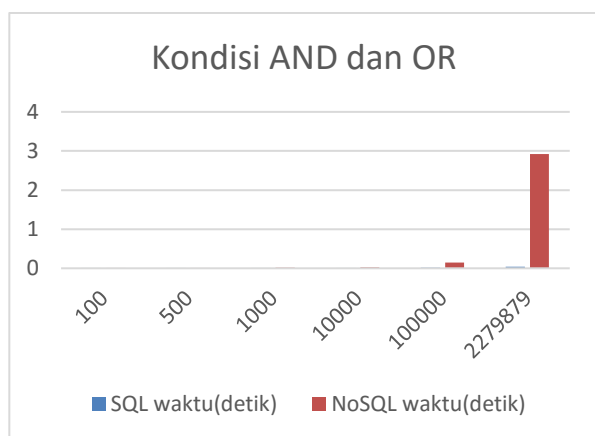
Grafik 6. Pencarian dengan kondisi AND

7. Pengujian ketujuh kombinasi pencarian AND dan OR

Tabel 7. Pencarian kondisi AND dan OR

No.	Jenis	Sintaks	Jumlah Data	Waktu (detik)
1	SQL	SELECT * FROM sales.dbo.T100 WHERE Category='1701100' AND (City='SHELDON' OR City='CORALVILLE')	100	0,001
			500	0,001
			1000	0,001
			10000	0,002
			100000	0,014
			2279879	0,038
2	NoSQL	db.getCollection("sales_2016").find({ "Category" : "1701100", "Sor" : [{ "City" : "SHELDON" }, { "City" : "CORALVILLE" }]}).explain("executionStats");	100	0,001
			500	0,001
			1000	0,016
			10000	0,023
			100000	0,145
			2279879	2,923

Pengujian ketujuh mencoba mencari data pada tabel kategori, city, zip_code, jika pencarian kategori yang didalamnya ada salah satu data terpenuhi antara city dan zip_code maka data akan tampil. Dan hasil pencarian ini waktu antara SQL cepat ketika memproses pada data besar dan NoSQL lebih lambat.



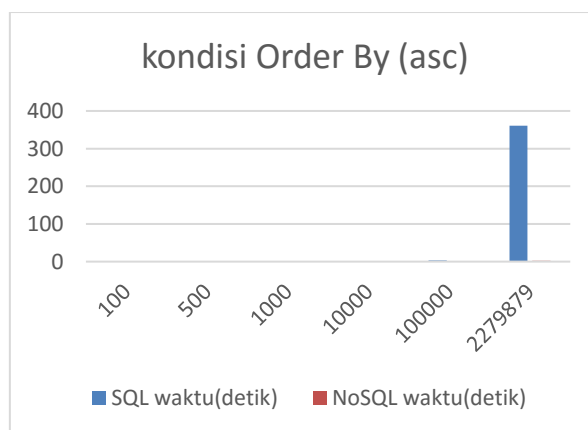
Grafik 7. Pencarian dengan kondisi AND dan OR

8. Pengujian kedelapan pencarian Order By (asc)

Tabel 8. Pencarian kondisi Order By (asc)

No.	Jenis	Sintaks	Jumlah Data	Waktu (detik)
1	SQL	SELECT * FROM sales.dbo.sales_2016 ORDER BY City ASC;	100	0,056
			500	0,113
			1000	0,130
			10000	0,349
			100000	2,699
			2279879	360,712
2	NoSQL	db.getCollection("T100000").find({}).sort({ "City" : NumberInt(1) }).explain("executionStats");	100	0,001
			500	0,001
			1000	0,001
			10000	0,011
			100000	0,101
			2279879	2,296

Pengujian delapan mencoba mencari data pada tabel city, jika pencarian data dari urutan atas maka data akan tampil. Dan hasil pencarian ini waktu antara SQL lebih lama dari pencarian NoSQL.



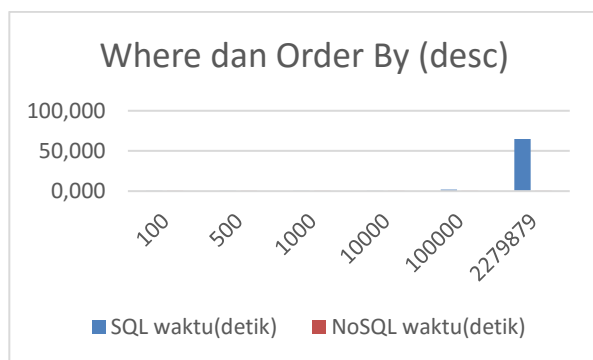
Grafik 8. Pencarian dengan kondisi Order By (asc)

9. Pengujian kesembilan pencarian kombinasi WHERE dan Order By (desc)

Tabel 9. Pencarian kondisi WHERE dan Order By (desc)

No.	Jenis	Sintaks	Jumlah Data	Waktu (detik)
1	SQL	SELECT * FROM sales.dbo.T1000000 WHERE Pack < 6 ORDER BY Invoice_Item_Number DESC	100	0,060
			500	0,103
			1000	0,108
			10000	0,296
			100000	2,046
			2279879	64,871
2	NoSQL	db.getCollection("sales_2016").find({ "Pack" : { "\$ne" : NumberLong(6) } }).sort({ "Invoice_Item_Number" : NumberInt(-1) }).explain("executionStats");	100	0,001
			500	0,001
			1000	0,002
			10000	0,039
			100000	0,257
			2279879	0,324

Pengujian kesembilan mencoba mencari data dengan kondisi where dan diurutkan dari terkecil tabel invoice_number maka data akan tampil. Dan hasil pencarian ini waktu antara SQL proses pencarian data lebih lama dari NoSQL.



Grafik 9. Pencarian dengan kondisi Where dan Order By (desc)

4. KESIMPULAN

Berdasarkan hasil percobaan pencarian data dengan query tampil seluruh data, salah satu data, AND, OR, AND dan OR, Order By (asc), dan Order By (desc) yang dilakukan dapat dilihat pencarian dengan pengujian kecepatan pencarian berdasarkan waktu, ada beberapa query yang dimana pencarian SQL lebih cepat pada pengujian keenam dan kelima dengan perbedaan waktu 0,1 detik dan 0,8 detik. Disisi lain NoSQL tetap proses query pencarian data lebih cepat dari SQL.

Banyaknya sintaks pencarian dalam sebuah data bisa disajikan lebih banyak dalam penelitian selanjutnya.

DAFTAR PUSTAKA

- [1] A. AAN SAID DAROINI, "PERBANDINGAN PENGGUNAAN NOSQL MONGODB DAN MYSQL PADA BASIS DATA FORUM KOMUNIKASI," *J. Manaj. Inform.*, 2016.
- [2] K. Agarwal, Sarthak and Rajan, "Analyzing the performance of NoSQL vs. SQL databases for Spatial and Aggregate queries," *Free Open Source Softw. Geospatial Conf. Proc.*, 2017, doi: <https://doi.org/10.7275/R5736P26>.
- [3] P. P. Srivastava, S. Goyal, and A. Kumar, "Analysis of various NoSql database," 2016, doi: 10.1109/ICGCIOT.2015.7380523.
- [4] S. Venkatraman, K. F. S. Kaspi, and R. Venkatraman, "SQL Versus NoSQL Movement with Big Data Analytics," *Int. J. Inf. Technol. Comput. Sci.*, 2016, doi: 10.5815/ijitcs.2016.12.07.
- [5] Y. L. Choi, W. S. Jeon, and S. H. Yoon, "Improving database system performance by applying NoSQL," *J. Inf. Process. Syst.*, 2014, doi: 10.3745/JIPS.04.0006.
- [6] N. A. Rakhmawati, M. Zuhri, R. P. Wibowo, A. Romadhon, H. Ardiansyah, and O. Khoirul, "Benchmarking MySQL and NoSQL Databases on Egovbench Application," *J. Inf. Technol. Its Util.*, 2019, doi: 10.30818/jitu.2.1.2080.
- [7] C. Bazar, C., & Losif, "The Transition from RDBMS to NoSQL. A Comparative Analysis

of Three Popular Non-Relational Solutions: Cassandra, MongoDB and Couchbase," *Database Syst. J.*, 2014.

- [8] A. K. Suman and M. Gyanchandani, "Improved Performance of Hive Using Index-Based Operation on Big Data," in *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2018, pp. 1974–1978, doi: 10.1109/ICCONS.2018.8663095.
- [9] A. Parssian, W. Yeoh, and M. S. Ee, "Quality-Based SQL: Specifying Information Quality in Relational Database Queries," *Computer (Long. Beach. Calif.)*, vol. 48, no. 9, pp. 69–74, 2015, doi: 10.1109/MC.2015.264.
- [10] S. S. Nyati, S. Pawar, and R. Ingle, "Performance evaluation of unstructured NoSQL data over distributed framework," 2013, doi: 10.1109/ICACCI.2013.6637424.
- [11] M. Silalahi, "PERBANDINGAN PERFORMANSI DATABASE MONGODB DAN MYSQL DALAM APLIKASI FILE MULTIMEDIA BERBASIS WEB," *Comput. Based Inf. Syst. J.*, 2018, doi: 10.33884/cbis.v6i1.574.
- [12] J. S. Van Der Veen, B. Van Der Waaij, and R. J. Meijer, "Sensor data storage performance: SQL or NoSQL, physical or virtual," 2012, doi: 10.1109/CLOUD.2012.18.